Graffoo TikZ library

Eugeniu Costetchi

06/05/2020

Contents

Abstract

TikZ-Graffoo is a TikZ extension to draw Graffoo diagrams. Graffoo is a tool for drawing OWL ontologies using a simple and well defined visual notation. This library implements the Graffoo visual notation.

1 Introduction

Graffoo, a Graphical Framework for OWL Ontologies[1], is an open source tool that can be used to present the classes, properties and restrictions within OWL ontologies, or sub-sections of them, as clear and easy-to-understand diagrams.

This library implements the Graffoo graphical notation[2] as a set of TikZ styles and macros for increased usability. You can use it to draw OWL ontology fragments or RDF graphs in general. The basic components of Graffoo allow one to create classes, datatypes, annotation, data and object properties, individuals and ontologies.

[1] https://essepuntato.it/graffoo/
[2] https://essepuntato.it/graffoo/specification/

2 Usage

Once you have downloaded and installed the `tikz-graffoo` package, using it is a piece of cake:

```
\usepackage{tikz-graffoo}
```

Drawing Graffoo diagrams in TikZ can be done either (a) by using the node and path styles or (b) by using the macros provided by this package. The latter is recommended for ease and clarity, but the former may be chosen to gain higher control.

2.1 Macros

The macros for drawing the basic Graffoo components are as follows. An example for each of these elements is depicted in Figure1.

- **generic entity**
  `\gEntity{TikZ parameters}{Id}{Content text}`
- **class**
  `\gClass{TikZ parameters}{Id}{Content text}`
- **class restriction**
  `\gGlassRestriction{TikZ parameters}{Id}{Content text}`
- **datatype**
  `\gDatatype{TikZ parameters}{Id}{Content text}`
- **datatype restriction**
  `\gDatatypeRestriction{TikZ parameters}{Id}{Content text}`
- **instance**
  `\gInstance{TikZ parameters}{position}{Id}{Content text}`
  The position parameter indicates where to situate the circle: `below`, `above`, `left` or `right` of the text.
- **literal value**
  `\gLiteral{TikZ parameters}{Id}{Content text}`
- **additional axioms**
  `\gAxiom{TikZ parameters}{Id}{Content text}`
- **prefix definitions**
  `\gPrefixes{TikZ parameters}{Id}{Content text}`

The LaTeXcode for the Figure 1 is the following.

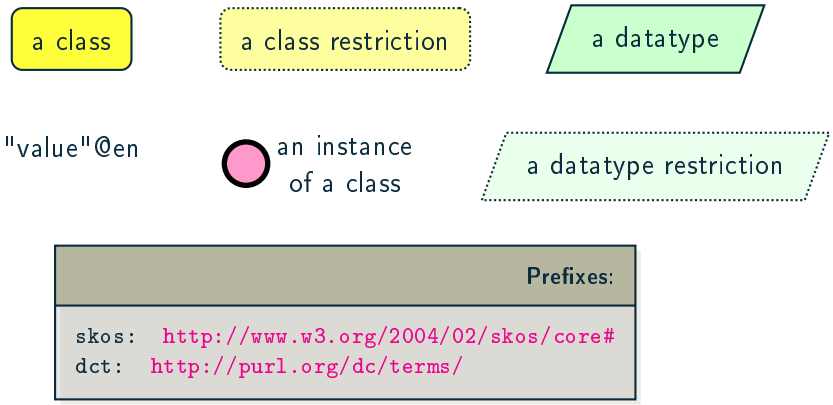

Figure 1: Drawing main Graffoo elements using macros

```
\begin{tikzpicture}
\gClass{}{class}{a class};
\gGlassRestriction{right=3em of class}
{class-restriction}{a class restriction};

\gDatatype{right=3em of class-restriction}
{datatype}{a datatype};

\gDatatypeRestriction{below=2em of datatype}
{datatype-restriction}{a datatype restriction};

\gLiteral{below=2em of class}
{literal}{\valueLang[en]{value}};

\gInstance{below=2em of class-restriction}
{left}{instance1}{an instance\\of a class};

\gPrefixes{below=6em of class-restriction}
{prefixes1}{
skos: \url{http://www.w3.org/2004/02/skos/core\#}\\
dct: \url{http://purl.org/dc/terms/}};
\end{tikzpicture}
```

The macros for drawing the basic Graffoo connectors are as follows. An example for each of these connectors is depicted in Figure2.

- **generic predicate**
  `\gPredicate{subject node}{object node}{predicate}`
- **object property**
  `\gObjectProperty{subject node}{object node}{predicate}`

- **datatype property**
  `\gDataProperty{subject node}{object node}{predicate}`
- **annotation property**
  `\gAnnotationProperty{subject node}{object node}{predicate}`
- **simple link**
  `\gLink{subject node}{object node}`

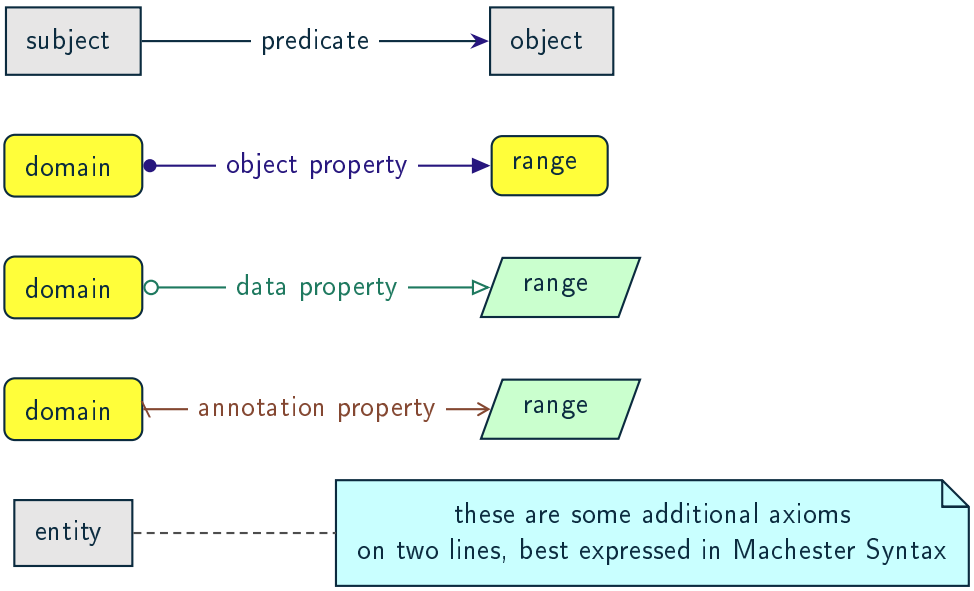

Figure 2: Drawing main Graffoo connectors using macros

The LaTeXcode for the Figure 2 is the following.

```
\begin{tikzpicture}
\gEntity{below=2em of literal}{s1}{ subject };
\gEntity{right=12em of s1}{o1}{ object };
\gPredicate{s1}{o1}{predicate};

\gClass{below=2em of s1}{s2}{ domain };
\gClass{right=12em of s2}{o2}{ range };
\gObjectProperty{s2}{o2}{object property};

\gClass{below=2em of s2}{s3}{ domain };
\gDatatype{right=12em of s3}{o3}{ range };
\gDataProperty{s3}{o3}{data property};

\gClass{below=2em of s3}{s4}{ domain };
```

```
\gDatatype{right=12em of s4}{o4}{ range };
\gAnnotationProperty{s4}{o4}{annotation property};

\gEntity{below=2em of s4}{s5}{ entity };
\gAxiom{right=7em of s5}{o5}{these are some additional axioms\\
on two lines, best expressed in Machester Syntax };
\gLink{s5}{o5};
\end{tikzpicture}
```

2.2 Styles

By using TikZ styles provided by this library, the same effect can be achieved as in the case of employing the macros listed above. The styles corresponding to each Graffoo element that should be applied to TikZ nodes are as follows.

- `entity`
- `class`
- `class-restriction`
- `datatype`
- `datatype-restriction`
- `literal` or `value` (they are equivalent)
- `instance-r`, `instance-l`, `instance-a`, `instance-r`
- `axiom`
- `pref` (should be applied to a node split into two parts)

The styles corresponding to each Graffoo property type that should be applied to TikZ paths are as follows.

- `predicate`
- `object-property`
- `data-property`
- `annotation-property`
- `link`

An example of drawing the diagram content as in Figure 1 and 2 using styles alone (no macros) is provided below.

```
\begin{tikzpicture}
\node[class] (class) {a class};

\node[class-restriction, right=2em of class]
(class-restriction) {a class restriction};

\node[datatype, right=2em of class-restriction]
(datatype) {a datatype};

\node[datatype-restriction, below=2em of datatype]
(datatype-restriction) {a datatype restriction};

\node[literal, below=2em of class]
(literal) { \valueLang[en]{value} };

\node[instance-l, below=2em of class-restriction]
(instance1) { an instance\\ of a class};

\node[entity, below=2em of literal]
(s1) { subject entity };
\node[entity, right=12em of s1]
(o1) { object entity };
\draw[predicate] (s1) -- (o1)
node[midway,fill=white,] {property};

\node[entity, below=2em of s1]
(s2) { subject entity };
\node[entity, right=12em of s2]
(o2) { object entity };
\draw[object-property] (s2) -- (o2)
node[midway,fill=white,] {object\\property};

\node[entity, below=2em of s2]
(s3) { subject entity };
\node[entity, right=12em of s3]
(o3) { object entity };
\draw[data-property] (s3) -- (o3)
node[midway,fill=white,] {data\\property};

\node[entity, below=2em of s3]
(s4) { subject entity };
\node[entity, right=12em of s4]
(o4) { object entity };
\draw[annotation-property] (s4) -- (o4)
```

```
node[midway,fill=white,] {annotation\\property};

\node[entity, below=2em of s4]
(s5) { subject entity };
\node[entity, right=12em of s5]
(o5) { object entity };
\draw[link] (s5) -- (o5);

\node [pref, below=3em of s5.west, anchor=north west]
(prefix) {\textbf{\sffamily Prefixes:}
\nodepart{two}
skos: \url{http://www.w3.org/2004/02/skos/core#}\\
dct: \url{http://purl.org/dc/terms/}};
\end{tikzpicture}
```

2.3 Helpers for literal values

Writing literal values using the Turtle or Manchester syntax is a little cumbersome. To make it easier we provide three macros.

- **plain literal value**
  `\valuePlain{value}` generates "value"^^xsd:string
- **typed literal**
  `\valueTyped[datatype]{value}` generates "value"^^datatype
  `\valueTyped{value}` generates "value"^^xsd:string
- **language tagged literal**
  `\valueLang[lang]{value}` generates "value"@lang
  `\valueLang{value}` generates "value"@en

3 License and contributions

`tikz-graffoo` may be distributed and/or modified under the conditions of the LaTeX Project Public License version 1.3 or later. The Current Maintainer of this work is Eugeniu Costetchi (costezki.eugen@gmail.com), who welcomes contributions to the package on GitHub.

The goal of this library is to provide a toolkit to draw quickly and efficiently beautiful OWL diagrams. Building such a library can be challenging for one pair of hands alone. If you have used this library yourself or have seen it used, we would greatly appreciate your opinion and feedback . Please open a Github issue or privately email the Current Maintainer with as much information as you can.

We also welcome technical feedback and bug reports in the form of GitHub Issues and pull requests.